

**AMENDMENTS TO THE CLAIMS**

Please amend claims 8, 10 and 18 as follows:

1. (Previously Presented) A method for exchanging data messages between a first block having a first protocol for exchanging messages and a second block having a second protocol for exchanging messages, the first protocol including a data type having a first structure, the second protocol including a data type having a second structure different than the first structure, the method comprising:

receiving a first representation, representing the first protocol, said first representation using regular expressions;

receiving a second representation, representing the second protocol, said second representation using regular expressions;

generating a first finite automaton for said first representation;

generating a second finite automaton for said second representation; and

automatically synthesizing an interface between the structurally different first and second protocols based on the first and second finite automatons.

2. (Previously Presented) The method of claim 1, further comprising:

automatically corresponding data from said structurally different first and second protocols.

3. (Previously Presented) The method of claim 2, further comprising:  
  
automatically translating data between said first protocol to said second protocol,  
said data in said first protocol having a first sequence, said data in said second protocol  
having a second sequence that is different from said first sequence.
  
4. (Previously Presented) The method of claim 2, wherein said generating a first finite  
automaton comprises:  
  
identifying the initial state of the first protocol;  
  
identifying a first sequence of data according to the first protocol;  
  
constructing derivatives of regular expressions; and  
  
eliminating equivalent expressions.
  
5. (Previously Presented) The method of claim 4, wherein said identifying a first sequence  
of data comprises:  
  
collecting data that is transferred during one or more transitions; and  
  
integrating said data with previous transitions.
  
6. (Previously Presented) The method of claim 5, further comprising:  
  
automatically translating data between said first protocol to said second protocol,  
said data in said first protocol having the first sequence, said data in said second protocol  
having a second sequence that is different from said first sequence.

7. (Previously Presented) The method of claim 1, further comprising:  
automatically translating data between said first protocol to said second protocol, said data in said first protocol having a first sequence, said data in said second protocol having a second sequence that is different from said first sequence
8. (Currently Amended) The method of claim 21, wherein automatically generating a third representation comprises:  
(a) selecting the interface state representing a first finite automaton state and a second finite automaton state;  
(b) identifying all outgoing transitions in said selected state;  
(c) determining a new state for each outgoing transition; and  
(d) repeating steps (a)-(c) for each interface state.
9. (Previously Presented) The method of claim 8, wherein generating a third representation comprises:  
identifying said permitted operations as operations that do not result in a data inconsistency.
10. (Currently Amended) The method of claim 8, further comprising ~~wherein said eliminating comprises:~~  
identifying non-deterministic transitions for each interface state; and

selecting a single outgoing transition for each interface state for each input value based upon priority parameters to generate a deterministic interface between the first and second protocols.

11. (Previously Presented) The method of claim 1, wherein said generating a first finite automaton comprises:

- identifying the initial state of the first protocol;
- identifying a first sequence of data according to the first protocol;
- constructing derivatives of regular expressions; and
- eliminating equivalent expressions.

12. (Previously Presented) A computer based system for exchanging data messages between a first block having a first protocol for exchanging messages and a second block having a second protocol for exchanging messages, the first protocol including a data type having a first structure, the second protocol including a data type having a second structure different than the first structure, the system comprising:

- storage device to store data and sequences of operations;
- a processor to receive signals from said storage device and to execute said sequences of operations;
- a receiving unit to transmit signals to said processor and to receive a first and second representation, representing the first and second protocols, said first and second representations using regular expressions;

an automata unit to generate a first finite automaton for said first representation and to generate a second finite automaton for said second representation; and

a synthesizing unit to automatically synthesize an interface between the structurally different first and second protocols based on the first and second finite automata.

13. (Previously Presented) The system of claim 12, further comprising:

a corresponding unit to receive signals from said processor and to automatically correspond data from said first and said second protocol, wherein said first protocol and said second protocol are structurally different

14. (Previously Presented) The system of claim 13, further comprising:

a translation unit to automatically translate data between said first protocol and said second protocol, said data in said first protocol having a first sequence, said data in said second protocol having a second sequence that is different from said first sequence.

15. (Previously Presented) The system of claim 13, wherein said automata unit comprises:

a first identifying unit to identify the initial state of the first protocol;

a second identifying unit to identify a first sequence of data according to the first protocol;

a derivative unit to construct derivatives of regular expressions; and

an eliminating unit to eliminate equivalent expressions.

16. (Previously Presented) The system of claim 15, wherein said second identifying unit comprises:

a data collection unit to collect data that is transferred as one or more transitions;

and

a data analyzer to integrate said data with previous transitions.

17. (Previously Presented) The system of claim 12, further comprising:

a translation unit to automatically translate data between said first protocol and said second protocol, said data in said first protocol having a first sequence, said data in said second protocol having a second sequence that is different from said first sequence.

18. (Currently Amended) The system of claim 12, wherein the product unit comprises:

a selection unit to select an interface state representing a first finite automaton state and a second finite automaton state;

an identifying unit to identify outgoing transitions in said selected state; and

a state unit to determine a new state for each outgoing transition;

19. (Previously Presented) The system of claim 18, wherein the product unit further comprises:

a consistency unit to identify said permitted operations as operations that do not result in a data inconsistency.

20. (Previously Presented) A computer readable medium storing instructions which, when executed by a processing system, cause the system to perform a method for exchanging data messages between a first block having a first protocol for exchanging messages and a second block having a second protocol for exchanging messages, the method comprising:
- receiving a first representation of the first protocol;
  - receiving a second representation of the second protocol;
  - generating a first finite automaton for said first representation;
  - generating a second finite automaton for said second representation;
  - generating a third representation of one or more permitted operations of said first and second finite automata; and
  - automatically eliminating non-determinisms in said third representation.
21. (Previously Presented) The method of claim 1, further comprising:
- automatically generating a third representation, representing one or more permitted operations of said first and second finite automata.
22. (Previously Presented) A method for exchanging data messages between a first block having a first protocol and a second block having a second protocol, the method comprising:
- generating a first finite automaton corresponding to the first protocol;
  - generating a second finite automaton corresponding to the second protocol;
  - generating a representation of one or more permitted operations of the first and second finite automata; and
  - automatically eliminating at least one non-determinism in the representation.